Scrum Artifacts

> These artifacts are essential tools used in Agile project management to plan, track, and deliver work efficiently.

We use a ride-sharing app like Uber) to maintain consistency and clarity:

Title Page

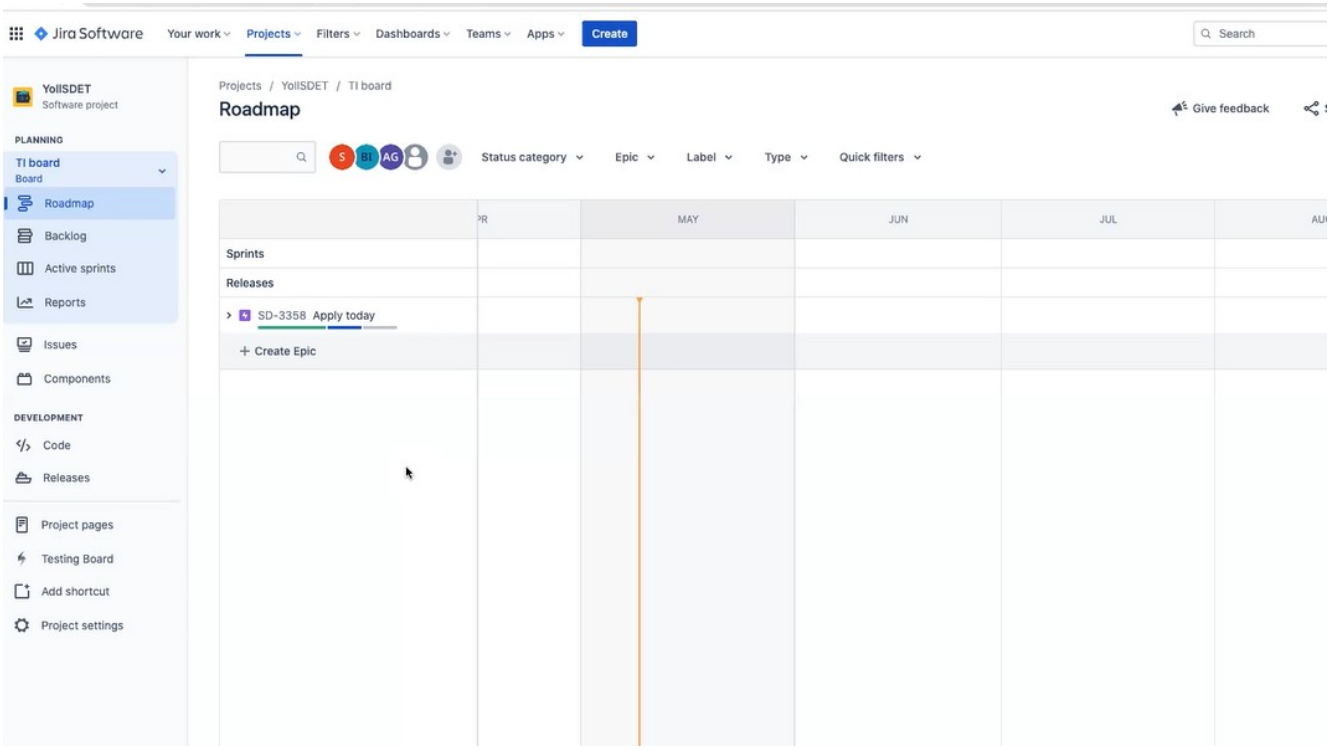
- **Explanation**: This page introduces the document, which focuses on **Scrum Artifacts**. These artifacts are essential tools used in Agile project management to plan, track, and deliver work efficiently.
- **Example**: Imagine you're working on a ride-sharing app like Uber. The Scrum artifacts (Roadmap, Product Backlog, Sprint Backlog, etc.) will help you organize and prioritize features like *real-time ride tracking*, *payment integration*, and *driver ratings*.

Scrum Artifacts Overview

- **Roadmap**, **Product Backlog**, **Sprint Backlog**, and **Monitoring Progress Toward Goals**: These tools help teams stay aligned and deliver value incrementally.
- **Example**: For the Uber-like app, the **Roadmap** outlines high-level goals like *launching the app*, *adding real-time tracking*, and *integrating payment systems*. The **Product Backlog** lists specific features like *user registration*, *ride booking*, and *driver ratings*. The **Sprint Backlog** focuses on tasks for the current Sprint, such as *designing the ride-booking interface* or *implementing payment processing*.

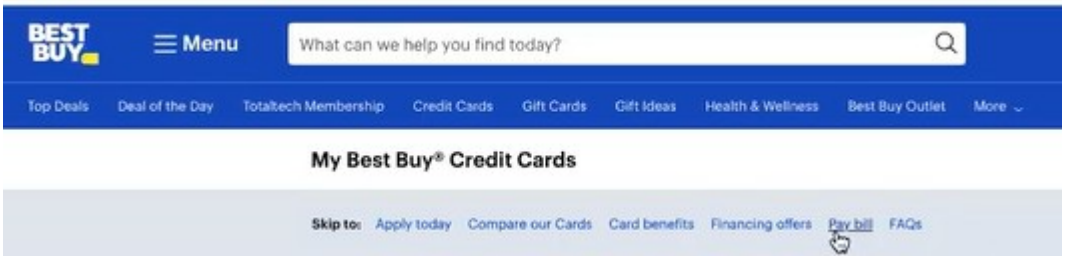
Roadmap

- **Explanation**: The **Roadmap** is a high-level plan that breaks down into **epics** and **user stories**. Epics are large pieces of work (e.g., *Ride Booking*), and user stories describe specific features from the user's perspective.
- **Example**: For the ride-sharing app, the roadmap might include epics like *Ride Booking*, *Payment Processing*, and *Driver Management*. A user story under *Ride Booking* could be: *As a user, I want to book a ride so that I can get to my destination quickly.*



Epics:

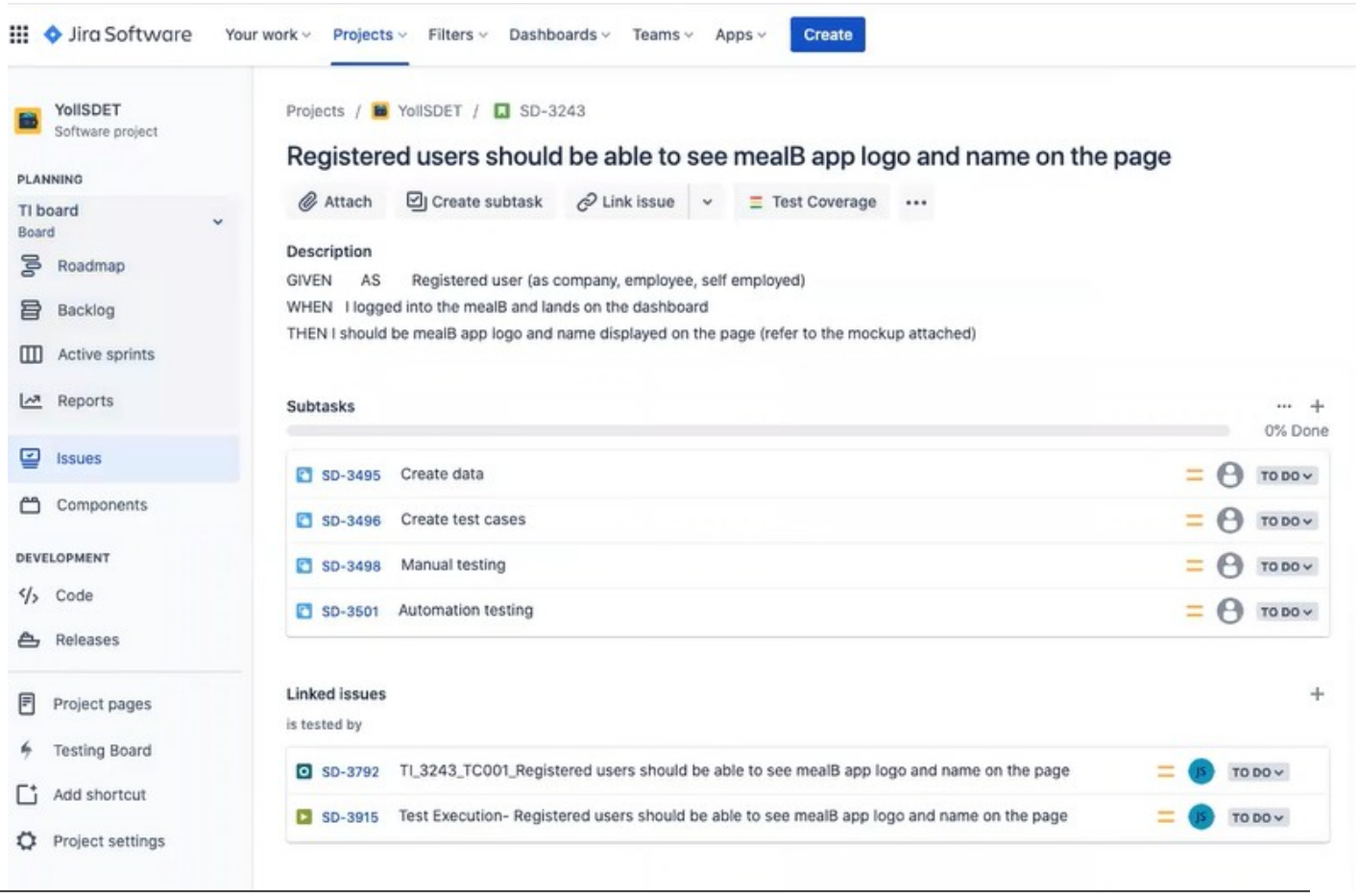
Pay bill is an **Epic**



User Stories: Online, By mail, By phone, By text, In store

Online
Make a payment online or check your balance.
Pay now*
By mail
Send your payment, including account number, to the address printed on your statement.
By phone
Call 1-888-574-1301. Have your account number and valid check from a U.S. bank ready when you call. There is no fee when you pay through our automated system.
By text
Text PAY to 81964 using the mobile device connected to your My Best Buy Credit Card account. After we confirm your identity, you'll receive a text asking how much you would like to pay and which payment account you would like to use. You'll receive a payment confirmation via text.
Message and Data rates may apply. Payment account information must be on BestBuy.AccountOnline.com to take advantage of pay by text — feature is not available for debit card payments.
In store
Our Blue Shirts can help you process your payment.
Find a store

Substasks:



Roadmap and Requirements Example

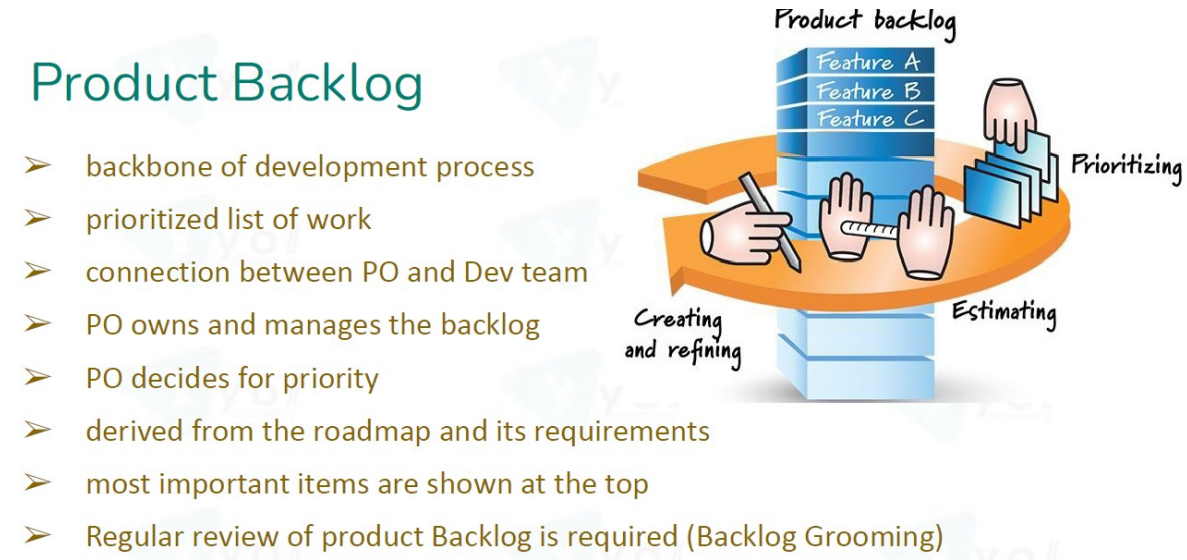
- **Explanation**: This page shows a sample roadmap for a product. It includes initiatives like *App Launch*, *Real-Time Tracking*, and *Payment Integration*, with a timeline from September to April.
- **Example**: For the Uber-like app, the roadmap might include initiatives like *App Launch* (September), *Real-Time Tracking* (November), and *Payment Integration* (January). Each initiative would have a timeline, such as *App Launch* being completed by October, followed by *Real-Time Tracking* in December.

Roadmap Breakdown into Epics and User Stories

- **Explanation**: The roadmap is broken into **epics** (e.g., *Ride Booking*), which are further divided into **user stories** (e.g., *Book a ride*). These user stories guide the development team in building specific features.
- **Example**: For the ride-sharing app, an epic like *Ride Booking* could include user stories such as:
    - *As a user, I want to book a ride so that I can get to my destination quickly.*

- *As a user, I want to see the estimated fare before booking so that I can plan my budget.*
- *As a user, I want to choose my preferred vehicle type (e.g., economy, premium) so that I can travel comfortably.*

Product Backlog

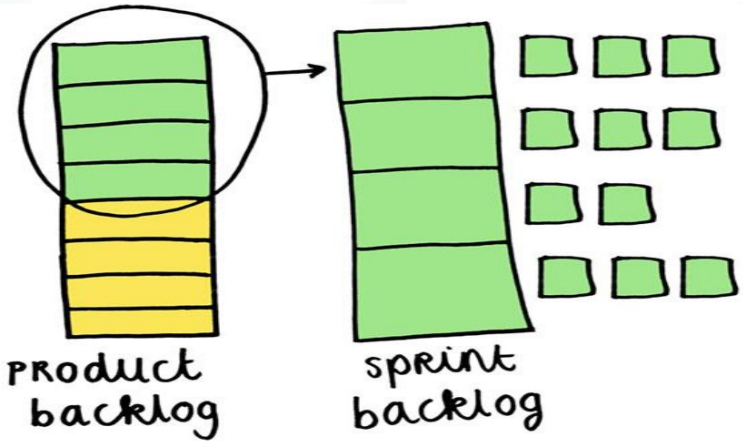


- **Explanation**: The **Product Backlog** is a prioritized list of work items derived from the roadmap. The **Product Owner (PO)** manages the backlog and ensures the most important items are at the top.
- **Example**: For the Uber-like app, the Product Backlog might include items like:
  - *High Priority*: Implement *Real-Time Ride Tracking*.
  - *Medium Priority*: Add *Multiple Payment Options* (credit card, PayPal, etc.).
  - *Low Priority*: Add a *Ride History* feature for users to view past trips.

Sprint Backlog

Sprint Backlog

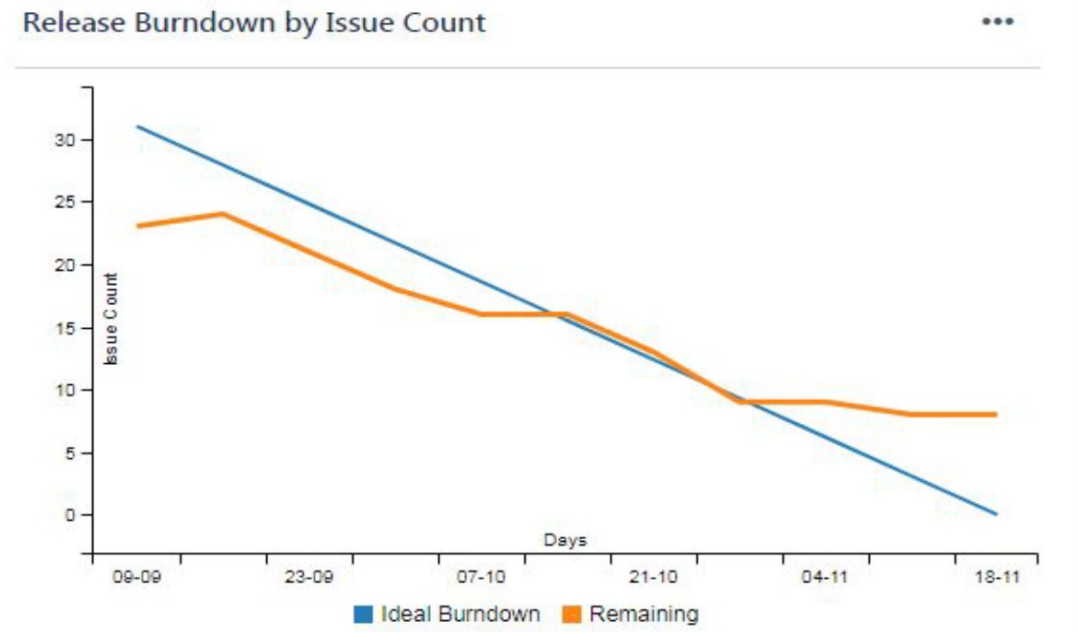
➢ set of Product Backlog items selected for the Sprint

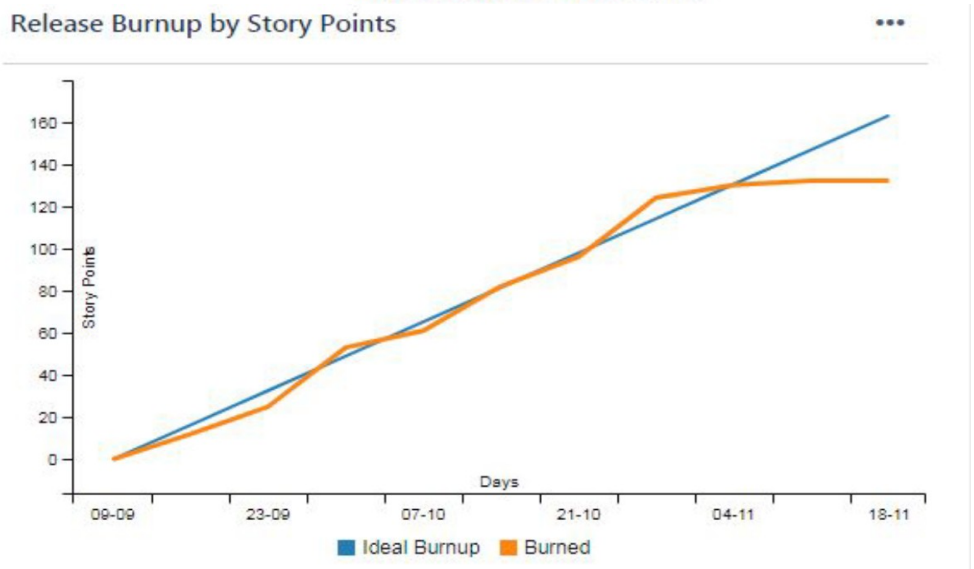


- **Explanation**: The **Sprint Backlog** is a subset of the Product Backlog, containing tasks the team commits to completing during the current Sprint.
- **Example**: If the Sprint goal is to improve the *Ride Booking* feature, the Sprint Backlog might include tasks like:
  - *Design the ride-booking interface.*
  - *Integrate the map API for location selection.*
  - *Add a fare estimation feature.*

Monitoring Progress Toward Goals

- **Explanation**: Teams use tools like **Burn-down Charts** and **Burn-up Charts** to track progress. The **Product Owner (PO)** reviews progress during Sprint Reviews to ensure the team is on track.
- **Example**: If a team is working on the *Real-Time Ride Tracking* feature, they might use a burn-down chart to track how many tasks are left. If the chart shows they're behind schedule, the team might adjust by re-prioritizing tasks or adding more resources.





Burn-down Charts and Burn-up Charts Explained with Examples for an Uber-like App

Both **Burn-down Charts** and **Burn-up Charts** are visual tools used in Agile project management to track progress during a Sprint or a project. They help teams understand how much work has been completed, how much is left, and whether they are on track to meet their goals. Below, I'll explain both charts in detail using the example of developing an **Uber-like ride-sharing app**.

---

1. Burn-down Chart

What is a Burn-down Chart?

A **Burn-down Chart** shows the amount of work remaining over time. It typically has:

- **X-axis**: Time (e.g., days or Sprints).
- **Y-axis**: Work remaining (e.g., story points, tasks, or issues).
- **Ideal Burn-down Line**: A straight line from the total work at the start to zero at the end of the Sprint.
- **Actual Burn-down Line**: A line that shows the real progress of the team.

How It Works:

- At the start of the Sprint, the team estimates the total work (e.g., 100 story points).
- Each day, the team updates the chart to reflect the work completed.
- The goal is for the "Actual Burn-down Line" to meet or stay below the "Ideal Burn-down Line," indicating the team is on track.

Example for an Uber-like App:

Let’s say the team is working on a Sprint to develop the **Ride Booking** feature for the app. The Sprint is 2 weeks long (10 working days), and the team has estimated **50 story points** of work.

- **Day 1**: The team completes 5 story points. The remaining work is 45.
- **Day 2**: The team completes another 5 story points. The remaining work is 40.
- **Day 5**: The team has completed 25 story points. The remaining work is 25.
- **Day 10**: The team completes all 50 story points. The remaining work is 0.

If the team is behind schedule (e.g., on Day 5, they’ve only completed 20 story points instead of 25), the Burn-down Chart will show the "Actual Burn-down Line" above the "Ideal Burn-down Line." This signals that the team needs to adjust their pace or scope to meet the Sprint goal.

---

2. Burn-up Chart

What is a Burn-up Chart?

A **Burn-up Chart** shows the amount of work completed over time, as well as the total scope of the project. It typically has:

- **X-axis**: Time (e.g., days or Sprints).
- **Y-axis**: Work completed (e.g., story points or tasks).
- **Total Scope Line**: Represents the total work in the project or Sprint.
- **Work Completed Line**: Shows the cumulative work completed over time.

How It Works:

- The "Total Scope Line" may increase if new work is added during the Sprint.
- The "Work Completed Line" shows how much work has been done.

- The goal is for the "Work Completed Line" to meet the "Total Scope Line" by the end of the Sprint.

Example for an Uber-like App:

Let’s say the team is working on the **Payment Integration** feature for the app. The Sprint is 2 weeks long (10 working days), and the initial scope is **40 story points**.

- **Day 1**: The team completes 4 story points. The total scope remains 40.
- **Day 3**: The team completes another 6 story points. The total scope increases to 45 because 5 new story points are added (e.g., a new requirement for supporting multiple payment methods).
- **Day 5**: The team has completed 20 story points. The total scope is now 50 due to additional requirements.
- **Day 10**: The team completes all 50 story points. The "Work Completed Line" meets the "Total Scope Line."

If the team is behind schedule (e.g., on Day 5, they’ve only completed 15 story points instead of 20), the Burn-up Chart will show a gap between the "Work Completed Line" and the "Total Scope Line." This indicates that the team needs to either increase their pace or reduce the scope to meet the Sprint goal.

Key Differences Between Burn-down and Burn-up Charts

Aspect	Burn-down Chart	Burn-up Chart
What it shows	Work remaining over time.	Work completed and total scope over time.
Focus	Tracks how much work is left.	Tracks how much work is done and scope changes.
Scope Changes	Not visible (only shows remaining work).	Visible (shows if scope increases or decreases).
Best for	Teams with a fixed scope.	Teams with a dynamic or changing scope.

Real-World Application for an Uber-like App

Scenario 1: Burn-down Chart for Ride Booking Feature

- **Sprint Goal**: Develop the **Ride Booking** feature (50 story points).
- **Day 1**: Team completes 5 story points (45 remaining).
- **Day 5**: Team completes 25 story points (25 remaining).
- **Day 10**: Team completes all 50 story points (0 remaining).

If the team is behind on Day 5 (e.g., only 20 story points completed), the Burn-down Chart will show the "Actual Burn-down Line" above the "Ideal Burn-down Line," prompting the team to take corrective action.

Scenario 2: Burn-up Chart for Payment Integration Feature

- **Sprint Goal**: Develop the **Payment Integration** feature (initial scope: 40 story points).
- **Day 1**: Team completes 4 story points (total scope: 40).
- **Day 3**: Team completes 6 more story points (total scope increases to 45 due to new requirements).
- **Day 5**: Team completes 20 story points (total scope: 50).
- **Day 10**: Team completes all 50 story points.

If the team is behind on Day 5 (e.g., only 15 story points completed), the Burn-up Chart will show a gap between the "Work Completed Line" and the "Total Scope Line," indicating the need for adjustments.

Why These Charts Matter for an Uber-like App

- **Transparency**: Both charts provide a clear visual representation of progress, helping the team and stakeholders stay informed.
- **Early Warning**: If the team is falling behind, the charts provide early warnings, allowing for timely adjustments.
- **Scope Management**: Burn-up Charts are particularly useful for projects like an Uber-like app, where requirements often evolve during development (e.g., adding new payment methods or ride options).

By using these charts, the team can ensure that features like **Ride Booking**, **Payment Integration**, and **Real-Time Tracking** are delivered on time and within scope, ultimately leading to a successful product launch.

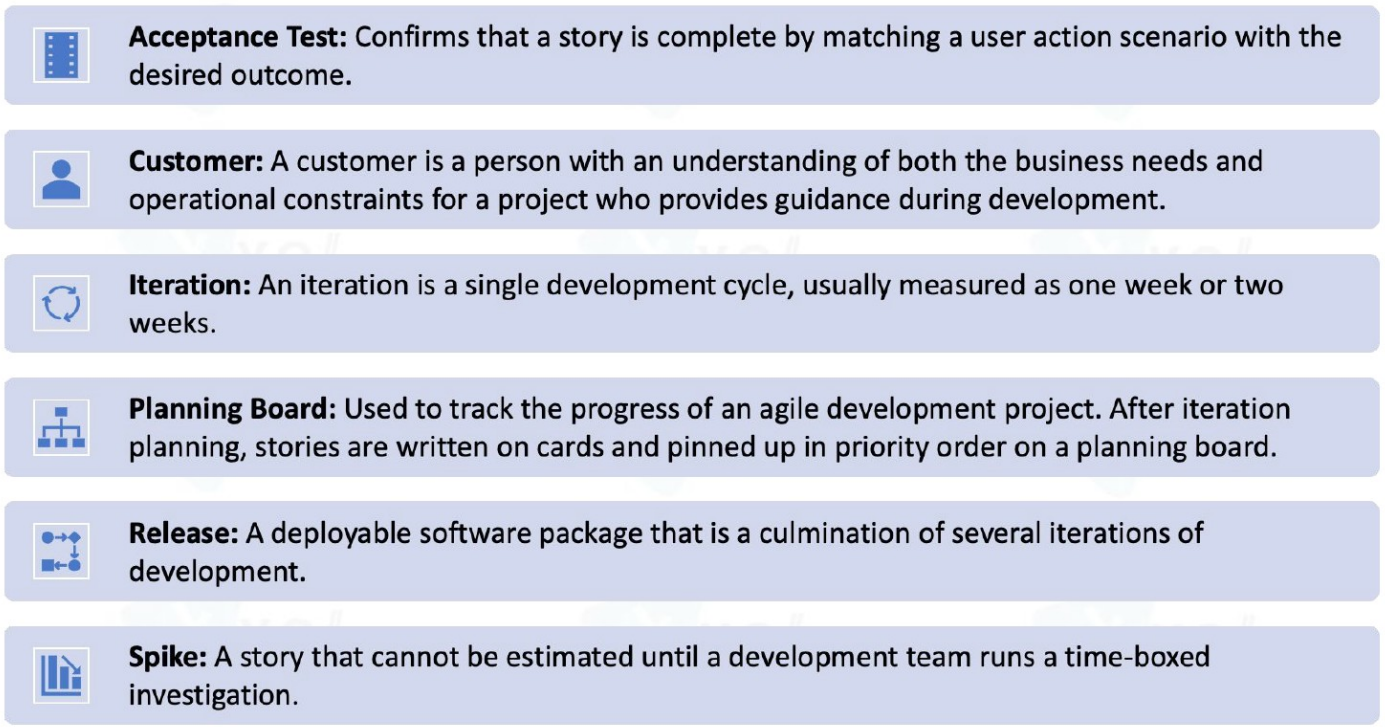
Release Burndown by Issue Count

- **Explanation**: A **Release Burndown Chart** shows the number of issues remaining over time, helping the team visualize progress toward a release.
- **Example**: If a team is developing the *Payment Integration* feature, the burndown chart might show 30 issues at the start of the Sprint. If only 10 issues are resolved by the halfway point, the team knows they need to speed up to meet the release deadline.

Release Burnup by Story Points

- **Explanation**: A **Release Burnup Chart** tracks the number of story points completed over time, showing how much work has been done and how much is left.
- **Example**: For the *Driver Management* feature, the burnup chart might show that 100 out of 150 story points have been completed. If the team is halfway through the Sprint, they're on track to finish on time.

Agile Glossary


Acceptance Test: Confirms that a story is complete by matching a user action scenario with the desired outcome.
Customer: A customer is a person with an understanding of both the business needs and operational constraints for a project who provides guidance during development.
Iteration: An iteration is a single development cycle, usually measured as one week or two weeks.
Planning Board: Used to track the progress of an agile development project. After iteration planning, stories are written on cards and pinned up in priority order on a planning board.
Release: A deployable software package that is a culmination of several iterations of development.
Spike: A story that cannot be estimated until a development team runs a time-boxed investigation.

Stand-up: A daily progress meeting (literally every stands up and meets to keep engaged and motivated), traditionally held within a development area

Story: A particular business need to be assigned to the software development team.

Timebox: A defined period of time during which a task must be accomplished.

Velocity: The budget of story units available for planning the next iteration of a development project. Velocity is based on measurements taken during previous iteration cycles.

Wiki: A wiki is a server program that allows users to collaborate in forming the content of a Web site.